From reproducibly analyzing Fitbit activity data to visualizing results.
JHSPH Biostat Qualifying Exam 2013 Take Home re-take

Abstract

We analyzed the number of steps taken by an individual for a period of two months in 2012 that was recorded using a Fitbit[2] device. Personal measurement devices have been on the rise and there is demand for new statistical methods to deal with this type of data. We estimated the average number of steps taken per day, inferred the average activity pattern within a day, and determined whether there was a difference in the activity pattern between weekdays and weekends. To do so, we explored the possibility of predicting missing observations. Finally, we built a Shiny[4] application that allows anyone to upload their own data and analyze with the methods implemented in this project.

This project is completely reproducible and all the code has been compiled in the `fitbitR` R package[1].

Introduction

Fitbit[2] is one of the popular devices on the market for collecting personal data such as the number of steps taken. Their devices can record activity data each minute and through the Fitbit API you can download your own activity data in different window intervals; for example every 5 minutes. Some enthusiasts have analyzed their own data[5] but questions remain such as what is the average number of steps taken per day, average activity patterns (within a day), and whether these patterns are different between weekends and weekdays. This project answers these questions and further allows users to analyze their own data through an application built with Shiny[4].

In particular, we have data from a single individual for a period of two months. As shown in Figure 1 (top) out of the 61 days, 8 are missing with no obvious difference between weekends and weekdays in the missing pattern. This individual regularly stops being active at 8pm on weekdays and 10pm on weekends, and then wakes up at 6 am on weekdays and 8 am on weekends as shown in Figure 1 (bottom).

Overall, there is indication of a difference in activity patterns between weekends and weekdays. Figure 1 (top) shows that the high activity peaks are more consistent on weekdays although they do change by date, versus the high peaks on the weekends which are more variable.

Methods and Results

Predicting missing observations

Four methods for predicting the missing observations have been implemented in `fitbitR`[1]. The first one, *overall-mean*, simply replaces the observations by the overall mean. The second one, *means*, replaces the missing observations by the mean from similar observations: those from the same interval and day of the week. The third one, *lm*, fits a linear regression model with a 10 degree of freedom natural spline on the Interval covariate, date and day of the week (dow). In other words using the following model:

$$Y = \beta_0 + \sum_{i=1}^{10} \beta_i ns_i + \beta_{date} + \beta_{dow}$$

The predicted values are truncated at 0 for any negative predictions (if any). The fourth method, *poisson*, fits a Poisson GLM using the same covariate structure as the *lm* method. The four methods, among others[1], were evaluated by training on 70% of the non-missing data and predicting on the remaining 30%. The error measure is the root mean square prediction error (RMSPE).

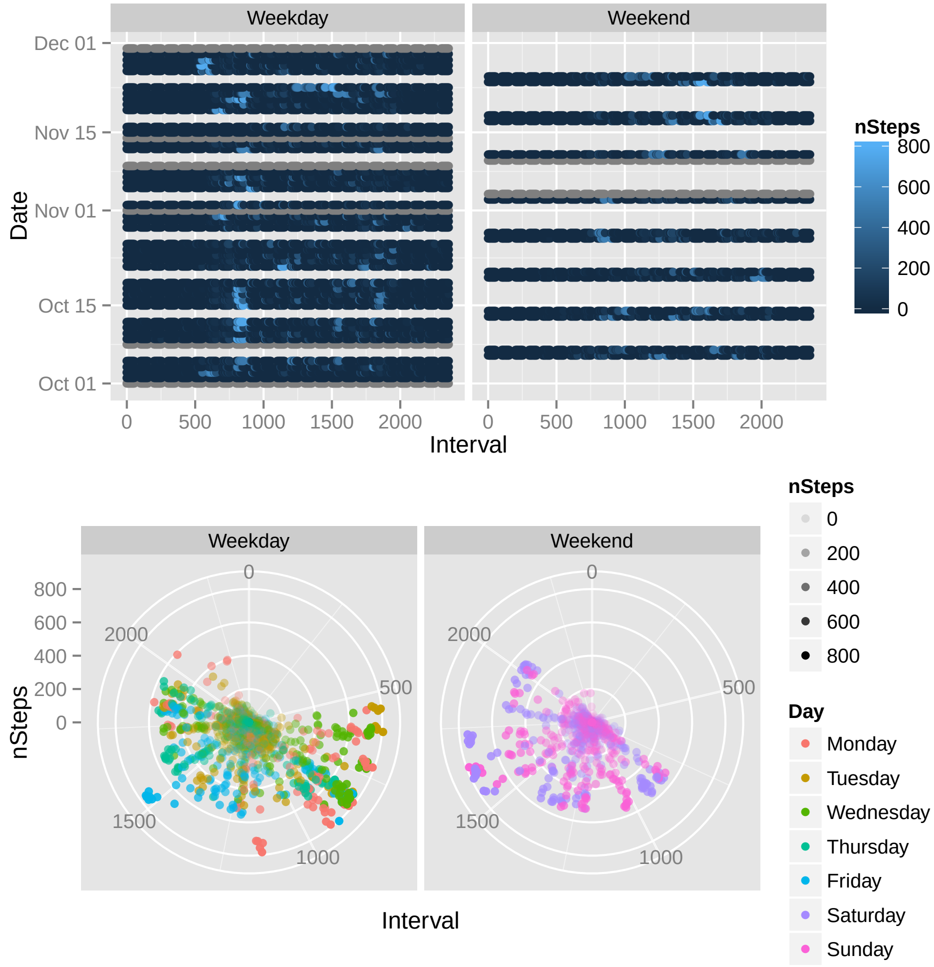

Figure 1: Exploratory plots of the number of steps (nSteps) for a specific individual along a two month period with data binned by 5 minute intervals. Top plot shows the data by Interval and Date separated by whether the day is a weekday or not. Missing observations (gray) are clearly visible. In addition, the activity peaks (light blue) are more consistent for weekdays while weekends seem more variable. Bottom plot shows the data in a in polar coordinates and we can notice that this individual regularly goes stops being active at 8pm and wakes at 6am on weekdays. On weekends, this individual wakes up at 8 am and has higher activity peaks on the afternoon.

overall-mean and *means* are straight-forward methods to predict the missing observations, but they heavily assume that the missing observations are at random and that there is a consistent pattern across similar observations (for *means*). *overall-mean* was used as a benchmark for poor-prediction.

lm was used despite the non-normality of the data due to it's robustness. We were not expecting a good result from this method, but it did out-perform *overall-mean* and *means*. *poisson* was used because the data are counts, despite overdispersion problems, and the flexibility of using natural splines. In both *lm* and *poisson*, 10 degrees of freedom on the natural splines worked better than

[1] Check reproduceAnalysis("pred") for more information.

using more.

We found that the *lm* and *poisson* methods were practically tied but did not improve the RMSPE by much: *overall-mean* 102.9 (SE 3.689), *means* 109.9 (SE 2.898), *lm* 99.7 (SE 3.283), and *poisson* 99.89 (SE 3.223). Surprisingly *means* was outperformed by *overall-mean*, which could an indicator that the error measure is sensible to outliers and a robust one should be used.

Average number of steps taken per day

Once the data is binned by day, the straightforward method to estimate the average number of steps taken by day is by calculating the sample mean. The problem with this method is that it ignores the correlation in the number of steps taken between day i and day $i + 1$. To deal with this structure, we fitted several ARIMA models[3] before choosing to use an ARIMA(0, 0, 1) [equivalent to MA(1)] model.

While the naive method has lower standard errors than when using the ARIMA(0, 0, 1) model[2], we argue that this is because the naive model underestimates the actual number of steps taken per day by assuming that the measurements are independent. The results are shown in Table 1 for the original data and the four prediction methods. Note that there were no significant differences (t-tests for the difference in sample mean, data not shown) between the original data and the predicted sets.

	Estimate	SE	95% CI:L	95% CI:U
original	10672.62	740.11	9188.14	12157.11
lm	10870.20	623.33	9623.79	12116.62
poisson	10846.05	624.37	9597.55	12094.55
means	10840.95	623.21	9594.77	12087.13
overall-mean	10786.71	625.12	9536.71	12036.71

Table 1: Estimated average number of steps taken per day using an ARIMA(0, 0, 1) model. The estimate, standard error and 95 percent confidence intervals (based on the t-distribution) are shown for the original data and for the four prediction methods.

Average activity pattern

To infer the average activity pattern we again compared two methods. The naive method is to take the mean of the observations grouped by interval. This method produces a highly variable estimate of the average activity pattern as shown in the appendix (Figure 3). It is thus important a smoother function in order to have a more interpretable picture of the average activity pattern.

For this purpose we fitted a General Additive Model (GAM) using the quasipoisson family and cubic regression splines. The quasipoisson family was chosen because the data is a set of counts with high overdispersion. The cubic regression splines are useful to generate a smooth curve that can be easier to interpret. Figure 2 (top) shows the overall activity pattern. From the result we can interpret that this person has a high peak of activity between 6 and 9 am. Then it stabilizes from 10 am to 4 pm. There is a slight increase at 5 and 6 pm and decreases until very low levels at 10 pm. The person is rarely active from 10 pm to 5 am.

[2] Check reproduceAnalysis("Q1") for more information.

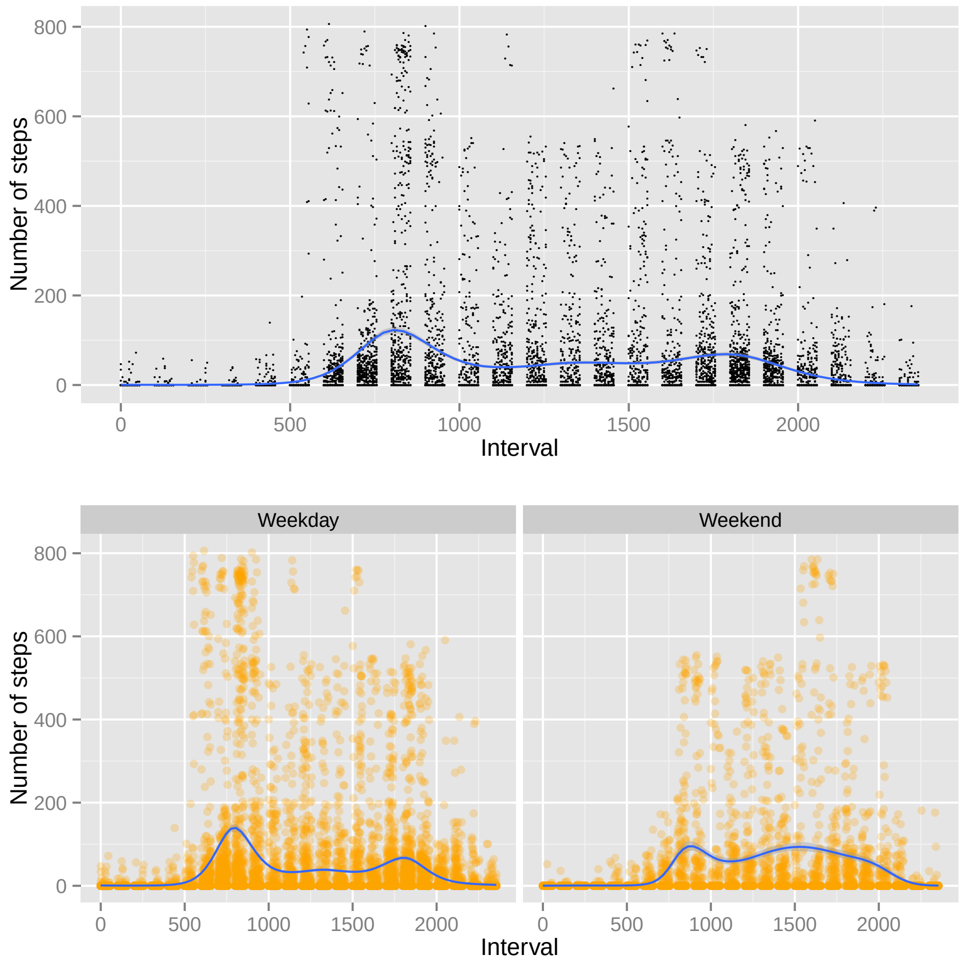

Figure 2: Average activity pattern over time (within a day). Top plot uses all the data while the bottom plot separates the data by whether it's a weekend or a weekday Blue curves are GAM models fitted for the quasipoisson family and with cubic regression splines.

Weekdays vs weekends

To answer the question of whether there is a difference in activity patterns between weekdays and weekends, we fit a GAM model using the quasipoisson family and cubic regression splines as described previously, but with the additional covariate of whether the observation corresponded to a weekday or weekend. The corresponding t value for the *Weekend* coefficient is 6.4752 with a p-value of 9.7566×10^{-11}. Therefore, there is a significant difference in the activity patterns between weekdays and weekends. This result was invariant to whether the original data or one with predicted values was used.

Figure 2 (bottom) we can notice how this individual has two activity peaks during weekdays as described previously. On weekends, this individual gets active later in the morning and keeps a rather similar activity level during the day (except for lunch time) and eventually stops being active later on the day compared to weekdays.

Conclusions

This individual takes an average of 10672 95% CI:(9188, 12157) steps per day, gets active early in the morning before stabilizing until his/her work is probably over at 5pm and goes home. During weekends, he/she gets active later during the day and overall keeps a higher level of activity on weekends.

The methods developed in this project should prove helpful to other individuals who want to analyze their Fitbit[2] activity data using the Shiny[4] application implemented in `fitbitR`[1]. The Shiny application includes the prediction methods although they did not affect the results in this current analysis. Thus, there is room for improving the prediction step.

References

[1] L. Collado-Torres. *fitbitR: JHSPH Biostat qualifying exam 2013 take home re-take by L. Collado-Torres*. R package version 0.2. URL: `https://github.com/lcolladotor/fitbitR`.

[2] Fitbit Inc. *fitbit*. URL: `http://fitbit.com` (visited on 06/06/2013).

[3] R. Hyndman and Y. Khandakar. “Automatic time series forecasting: The forecast package for R”. In: *Journal of Statistical Software* (2008).

[4] RStudio and Inc. *shiny: Web Application Framework for R*. R package version 0.7.0. 2013. URL: `http://CRAN.R-project.org/package=shiny`.

[5] J. T. Rubin. *One Full Year of FitBit Pedometer Data*. URL: `http://www.jamierubin.net/2013/03/20/one-full-year-of-fitbit-pedometer-data-part-1-a-look-back/` (visited on 06/04/2013).

A Average activity pattern: mean method

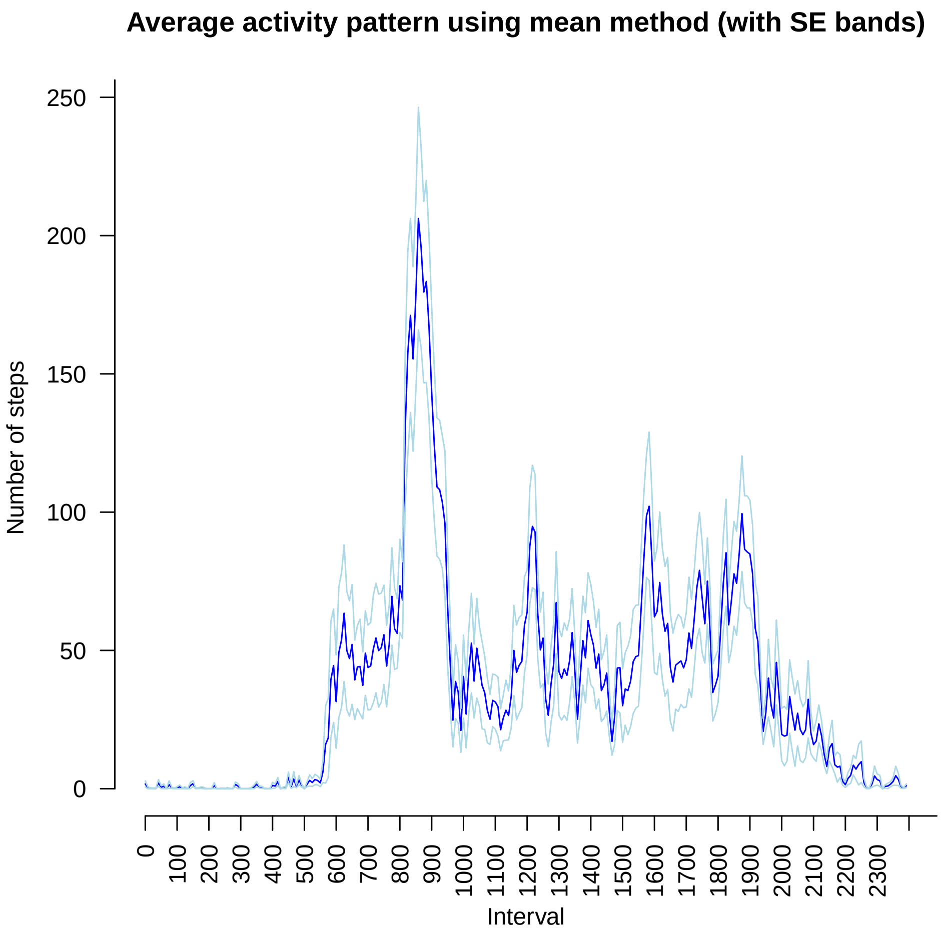

Figure 3: Average activity pattern over time (within a day) using the naive mean method.

B Running the Shiny application

To run the Shiny application, you just have to run the following commands from R. Sadly, we did not get an account in the Beta Shiny server by the due time despite our efforts to get one.

```
library(fitbitR)
fitbitShine()
```

C Reproducibility

Please check `https://github.com/russojhsph/fitbitR` for details on how to install the *fitbitR* package and reproduce the results including this report.

Note that `reproduceAnalysis('all')` takes around 3 minutes to complete.